Entropy-Weighted Random Forests: Robust Learning in Noisy, Imbalanced Data

Anusha Agarwal and Viola Pande

01/20/2026 | Yilmaz Period 4 | Quarter 2 Project

Abstract

Random Forests are a highly applicable machine learning approach to classification problems within the biomedical and chemical industries, be it genomics, protein science, disease detection. However, datasets belonging to these domains are often ridden with noise and an overflow of features, simply due to the nature of biological data collection and measurement. With noisy and high-dimensional data, Random Forests tend to lose their robustness. For this reason, we propose a novel weighted voting system (EWRF) to the aggregation of predictions from decision trees, drawing an exponential relationship with the Shannon entropy of the final leaf node. We tested our weighting system across different maximum tree depths and found it to be most effective when paired with moderate depth constraint. We observed a consistent increase in both accuracy and F1-score, achieving up to 4.38% and 58.8% increases respectively. The input to our algorithm is features representing the presence/absence of specific structures in each chemical compound, and we use our EWRF to output a prediction of whether each compound is active (binding) or inactive. By improving upon the performance of Random Forests on noisy and imbalanced data, we are offering a breakthrough in biological machine learning, where data without noise is extremely uncommon.

Introduction

Individually, decision trees tend to overfit, continuing to split due to an overflow of possible features. This phenomenon leads to models that perform well on training data but poorly on testing data. The goal of the Random Forest algorithm (RF) is to combat this issue through ensemble learning, aggregating the predictions of many lower-performing decision trees to emit one high-level prediction. Conventionally, a majority voting system is applied to the aggregation of decision tree predictions, assigning each one the same priority. However, trees almost always differ in their prediction accuracies, which can actually be inferred directly from their final node entropy. Entropy refers to a tree node's disorder – the spread of instances at a node across different classes. A lower final node entropy indicates that the data has been split more precisely, leading to a higher confidence that the decision tree's prediction is correct.

By appending a weight to each tree's prediction based on their entropy prior to adding it to the RF compilation, we aim to utilize this relationship between disorder and accuracy. The input to our algorithm is a chemical compound. We then use a Random Forest to classify the instances as active or inactive based on their structural molecular features. Our weighting system is specifically tailored to handle noisy data, an area in which standard Random Forest fails. Removing features in the preprocessing stage poses the risk of accidentally discarding valuable information, hidden in subtleties that often appear in the biological realm. This is why our algorithm seeks to work through the noise rather than getting rid of it completely.

A proper weighting system applied to majority voting can provide a powerful boost in model reliability [1]. This is important because the Random Forest algorithm is extremely helpful in combatting the overfitting tendencies of decision trees, but in order to fully reap these benefits, their accuracy must be optimal.

Related Work

Several related studies have efforted to alter the voting system of Random Forest aggregation, including implementing a weighting system similar to our project. However, no one has investigated the effectiveness of entropy-based weights on RF accuracy.

A 2004 approach by Robnik-Šikonja [2] employed the performance of the trees on classifying similar instances as a measure of their overall contribution to the Random Forest, essentially giving a higher weight to the votes of decision trees that properly classified similar data. This weighting system increased accuracy on 11 of the 17 datasets tested. Evanthia et. al [3] investigated the same system (2% accuracy increase) as well as two others: weighting the trees based on metrics between databases (difference in training vs. test sets — 2% accuracy increase) and based on classification rates (1-2% accuracy increase). Rakesh Raushan attempted weighting votes by the decision trees' F1-scores, leading to an average increase in accuracy of around 1.25% [4]. Most recently, Zhang et. al [5] weighted predictions based on both classification rates and a probabilistic function [2] that calculated the trees' confidence, both of which led to no improvement in accuracy. However, when they attempted to combine these two factors, there was a jump in accuracy of 1.21%. They did this by using both the conventional accuracy metric and the probability intervals indicating the trees' confidence.

While these systems are innovative, our approach possesses the unique upside of examining a quality inherent to the trees — purity — to assess their value rather than relying on information derived from their outputs. As such, our system does not depend upon external testing data.

Dataset and Features

We chose DOROTHEA [6], which is a drug discovery dataset from the UC Irvine Machine Learning repository. It is a binary classification problem, where each chemical compound must be classified as active (a binding compound) or inactive. Our dataset is highly class-imbalanced, with around 90% of the data being classified as inactive, and only 10% of the data classified as active. In total, there are 1150 labeled instances, and a validation set of 350 instances was already provided to us as part of this. Besides this, we received a training set of 800 instances which we split into a training/testing set using an 80/20 split, as the final testing set for DOROTHEA was unlabeled, and any supervised learning algorithm like

Random Forest requires labeled data. The final training, validation, and test split was 640 training instances (55%), 350 validation instances (30%), and 160 testing instances (15%).

A key reason for choosing this dataset was the addition of 'probes', or distractor features, which hold no real predictive power for the model. These are deliberately added to the dataset to introduce challenges for supervised learning. This dataset has an equal number of predictive and probe features (50,000 each, 100,000 total), as it was designed for a feature selection challenge. This makes it an extremely high-dimensional and noisy dataset, which is ideal for evaluating algorithms that can handle this noise and irrelevance without relying on feature selection. As mentioned in the introduction, vanilla Random Forest often underperforms in these settings, as extraneous noise adds entropy to the leaf nodes and reduces the predictive power of the models. By including all 100,000 features in the development of our model, we are able to test it in a scenario relevant to real-world chemical and medical settings, as oftentimes in these situations feature selection is not feasible due to the possible unknown underlying biological mechanisms linking features. Specific features are not given names as once again, this was a feature selection algorithm development dataset, and the creators wanted to ensure no manual bias when selecting features.

We took a few main steps to preprocess the data. First, we ensured that there were no missing values. Then, we inspected the structure of the data. Each feature is binary, and represents either the presence or absence of specific chemical structures in each compound. Due to this, no normalization was necessary. We chose to not augment our data, as in chemical/medical datasets, creating synthetic structures could introduce correlations that aren't realistic. This dataset allows us to test Random Forest-based methods on extreme noise and feature conditions.

Methods

To begin, we trained a standard Random Forest using a scikit-learn implementation [8]. Random Forest is an ensemble learning model where many individual decision trees are trained, and their votes on the class of the sample are aggregated using majority voting for final classification. Each of these individual decision trees are tested on a bootstrap sample of the dataset created through double randomization, both in instance and feature selection. This allows the Random Forest to improve the generalization of regular decision trees. It is important to note that in regular implementations, each decision tree's vote is given an equal weight towards the final prediction for a sample. For the purposes of this study, we chose to test different depth constraints for our decision trees to understand their effects on performance in noisy high dimensional datasets like DOROTHEA.

We expand on these RFs by developing the Entropy-Weighted Random Forest (EWRF) which applies a voting system to each tree that weights the prediction of each tree based on their entropy. We used the standard Shannon entropy formula that is employed when training traditional decision tree models and determining their splits. The formula is given below, where p_i is the predicted probability of class *i* at the leaf node a sample reaches, and k is the total number of classes. Essentially, we can say that the lower the entropy, the more 'confident' a tree is with its prediction.

$$H = -\sum_{i=1}^{k} p_i * \log_2(p_i)$$

Figure 1. *Shannon entropy formula* [7]

After calculating individual entropies, we calculate entropy-based weights. The formula we developed is the exponential function $w = exp(-\alpha H)$, where α is a constant scaling factor (we used $\alpha = 10$) which magnifies the entropy difference between trees. This formula causes low entropies to have very high weights, and high entropies to be weighted much lower. We perform this calculation independently for each sample to ensure that the weights are locally adaptive.

$$P(y = c \mid x) = \frac{\sum_{t=1}^{T} w_t(x) \cdot p_t(c \mid x)}{\sum_{t=1}^{T} w_t(x)}$$

Figure 2. *Novel weighting formula*

At the final leaf node of each decision tree, a probability distribution for the possible classes is returned. We multiply each of these probabilities by the weights using the equation in Figure 2, where $p(c|x)$ is the probability predicted by tree t for class c, and $w(x)$ is the corresponding entropy-based weight. The output is a weighted probability distribution. This process is repeated with each decision tree in the Random Forest. Then, we add up all of the weights for each class and normalize them so that we get a final weighted probability distribution across the forest. From here, we can at last predict the class as the one with the higher probability.

This process allows us to penalize specific trees that have higher uncertainty than others, and also allows for adaptive sample weighting. Depending on the sample, certain trees may have lower entropies than others, which is why each sample defines new weights. This is extremely beneficial in datasets with noisy features, or complex class boundaries.

Experiments/Results/Discussion

To test the performance of our exponential entropy-weighted Random Forest voting, we conducted a number of experiments comparing our EWRF algorithm to

traditional Random Forests. The primary goal of these experiments was to systematically test whether the entropy-based adaptive voting could improve classification in highly noisy, class-imbalanced, and high-dimensional settings. In particular, we experimented with the number of decision trees and the maximum depth of the decision trees.

We experimented with RFs of both 50 and 100 trees, over a maximum depth of 5, 10, 15, 20, and no depth constraint. We chose these values to ensure that we were testing over a wide range of parameters, and exploring different types of trees ranging from shallow, high-bias to deep, high-variance trees. We ensured that all models were tested on the same data splits for consistency.

As the DOROTHEA dataset is extremely class-imbalanced, with almost 90% of samples belonging to the inactive class, the accuracy alone is not enough to assess performance. A model could have high accuracy while failing to identify any active, binding compounds. Therefore, for each of these experiments, we report the entropy-weighted testing accuracy, improvement in accuracy (Improvement(A)), the precision, the recall, the F1-score, and the improvement in F1-score (Improvement(F1)) [9]. We report our accuracy for context, but we acknowledge that this is a highly class-imbalanced problem. Therefore, we also report precision, recall, F1-score, and its improvement in order to provide a better understanding of how our model is learning, and making sure it isn't always just biased towards the majority class, which is a common pitfall of decision trees and Random Forests.

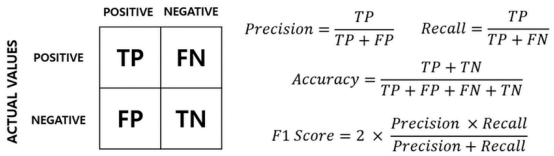

Figure 3. *Formulas for employed metrics* [10]

Overall, the entropy-weighted Random Forest voting demonstrated improvement over a number of experiments when compared to the traditional RF. We saw clear benefits in both accuracy and F1-score, with particularly large gains in the F1-score. The highest accuracy improvement was 4.38%, and the highest F1-score improvement was 58.82%. Below is a table reporting all metrics across all experiments, alongside graphs comparing the RF and EWRF accuracies and F1-scores across various tree depths for 50 trees.

To discuss the trends and patterns in the performance of our model, we can primarily look at the performance as a function of tree depth. At shallow depths, regular RF tends to produce highly uncertain predictions, producing trees with high-entropy leaf nodes that often default to the majority class to overwhelm the prediction. Our entropy weighting directly addresses this by exponentially down-weighting these types of trees, showing clear performance gains especially in the minority class which we can see in the high F1-score improvements at shallow depths in Figures 4 and 5.

Figure 4. *Metrics and confusion matrices for varied constraint depths (y) and number of trees (x)*

	n_trees=50	**Confusion Matrix:**	**100**	**Confusion Matrix:**
depth =5	Accuracy: 91.25% Improvement (A): 1.87% Precision: 0.5556 Recall: 0.6250 F1-score: 0.5882 Improvement (F): 58.82%	[136 8] [6 10]	Accuracy: 91.87% Improvement (A): 1.87% Precision: 0.5882 Recall: 0.6250 F1-score: 0.6061 Improvement (F): 49.49%	[137 5] [6 10]
10	Accuracy: 95% Improvement (A): 3.12% Precision: 0.7222 Recall: 0.8125 F1-score: 0.7647 Improvement (F): 32.99%	[139 5] [3 13]	Accuracy: 96.25% Improvement (A): 4.38% Precision: 0.8125 Recall: 0.8125 F1-score: 0.8125 Improvement (F): 37.77%	[141 3] [3 13]
15	Accuracy: 95.63% Improvement (A): 2.5% Precision: 0.8000 Recall: 0.7500 F1-score: 0.7742 Improvement (F): 18.16%	[141 3] [4 12]	Accuracy: 95% Improvement (A): 1.87% Precision: 0.7857 Recall: 0.6875 F1-score: 0.7333 Improvement (F): 17.33%	[141 3] [5 11]

20	Accuracy: 95% Improvement (A): 0% Precision: 0.7857 Recall: 0.6875 F1-score: 0.7333 Improvement (F): 0%	[141 3] [5 11]	Accuracy: 94.37% Improvement (A): 1.25% Precision: 0.7692 Recall: 0.6250 F1-score: 0.6897 Improvement (F): 9.71%	[141 3] [6 10]
None	Accuracy: 93.75% Improvement (A): 0% Precision: 0.7500 Recall: 0.5625 F1-score: 0.6429 Improvement (F): 0%	[141 3] [7 9]	Accuracy: 92.5% Improvement (A): 0% Precision: 0.7000 Recall: 0.4375 F1-score: 0.5385 Improvement (F): 0%	[141 3] [9 7]

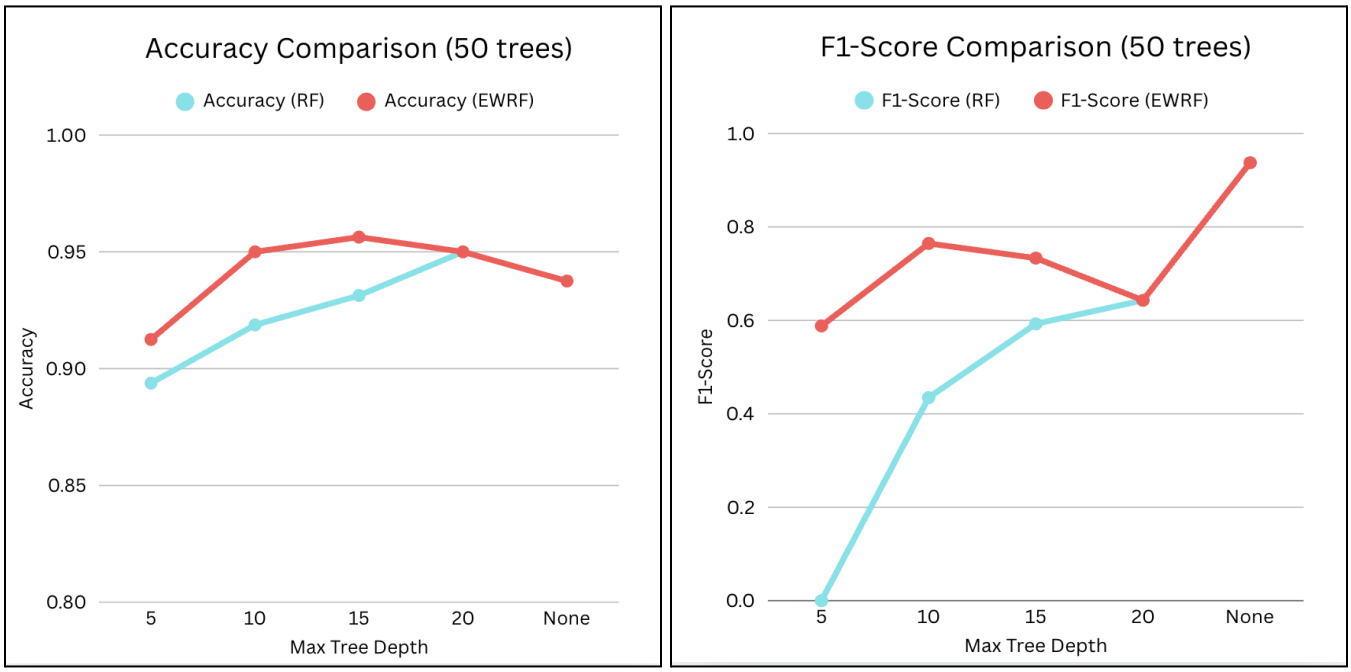

Figure 5. *Standard vs. entropy-weighted RF improvements in metrics*

However, as the tree depth increases further, (15 and beyond), we can see where our model's benefits begin to diminish. Overfitting is a concern both with high-dimensional data like DOROTHEA and Random Forest algorithms specifically. To mitigate this, we tested on different constrained tree depths and evaluated all the models on an entirely unseen testing dataset. However, deeper trees are more prone to overfitting, especially on noisy datasets like these. They may reach extremely confident/low-entropy predictions that are really based on noise. Our entropy-weighting model assumes that low entropy corresponds to trustworthy predictions, which breaks at these deeper splits, and it becomes less effective at distinguishing which trees have low entropy because they are meaningfully confident of if they have overconfidence from artificial noise. This explains why our benefits taper off when we reach higher depths. This indicates that EWRF may be most effective when it is combined with decision tree constraints and general model capacity control as we developed through our experiments. The optimal spot seems to fall around a maximum depth of 10, a moderate constraint value that allows the tree to utilize all the information it is given but avoid overuse to the point of overfitting.

As visible in the confusion matrices, the true negative rate of our algorithm is exceptionally high, which lies in agreement with the majority of our instances belonging to the negative class. This class imbalance is why our high improvement in F1-score is so important. This demonstrates that our entropy-weighted Random Forest is a substantially strong contender for tackling the class imbalance problem in Random Forests. It clearly shows that it is more accurately able to classify instances of the minority class when the traditional RF was not. Although significant, these jumps are reasonable, as on highly class-imbalanced datasets, extremely low initial F1-scores are likely due to poor performance in identifying the minority class, which seems to be the case with DOROTHEA and vanilla Random Forest.

These values are substantial in comparison to related work. While [2] established a high frequency of improvement, our approach aimed to optimize the *magnitude* of improvement, a goal within which we found substantial success. Our highest observed increase in accuracy was twice the jump in [3], [4], and [5]. However, as mentioned before, accuracy is not always a reliable metric in noisy and high-dimensional datasets. To address this, we can compare the precision improvement in [3], which was maximized at 8%, to our improvement in F1-score, which reached a maximum of almost 59%. While Evanthia et. al's spike in precision indicates their weighted voting system's ability to avoid false positives, our drastic F1-score increase demonstrates the EWRF's balanced gains across *both* precision and recall. In turn, our system provides more dramatic and comprehensive performance improvement for imbalanced and noisy classification tasks.

Conclusion / Future Work

Random Forests are a substantially robust algorithm when it comes to combatting the overall issue of overfitting within decision trees. The benefits of RF are widely applicable to countless domains, but dwindle when venturing into fields with data that tends to be higher in dimension and noise, such as biology. Our Entropy-Weighted Random Forest addresses this effectively by weighting the predictions of the less accurate decision trees based on their inherent final node entropy and applying this weighted probability distribution to the overall Random Forest vote. This outputs an, on average, more accurate and precise prediction, providing a groundbreaking new avenue for Random Forests to better learn and classify biological, medical, and chemical data. The EWRF performed best for moderately depth-constrained decision trees, solidifying our understanding of how entropy's usefulness diminishes as trees become either too simple or too complex.

Future work might study the effects of weighted voting with different models of entropy, such as Tsallis or von Neumann entropy. Additionally, researchers could combine our developed EWRF formula with other metrics, such as a hybrid weight derived from out-of-bag accuracy and entropy. Finally, this experiment could be expanded to test multiple classification problems, averaging the improvements in metrics over ten or more datasets for an overall increase in generalizability and reliability.

Contributions

Anusha: Drafted code. Analyzed dataset/features. Wrote methods. Analyzed experimental results. Synthesized presentation.

Viola: Debugged and ran code. Wrote abstract/intro/conclusion. Analyzed related work. Analyzed experimental results.

References

[1] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms. Taylor & Francis Group, LLC, p. 74. [Online]. Available: https://tjzhifei.github.io/links/EMFA.pdf

[2] M. Robnik-Šikonja, "Improving Random Forests," Machine Learning: ECML 2004, pp. 359–370, 2004, doi: https://doi.org/10.1007/978-3-540-30115-8_34.

[3] E. E. Tripoliti, D. I. Fotiadis, M. Argyropoulou, and G. Manis, "A six stage approach for the diagnosis of the Alzheimer's disease based on fMRI data," Journal of Biomedical Informatics, vol. 43, no. 2, pp. 307–320, Apr. 2010, doi: https://doi.org/10.1016/j.jbi.2009.10.004.

[4] R. Raushan, "Advanced Voting Method for Improving Random Forest Classification Algorithm Performance in Machine Learning," Imperial Journal of Interdisciplinary Research, vol. 3, no. 11, pp. 44–50, Nov. 2017.

[5] H. Zhang, B. Quost, and M.-H. Masson, "Cautious weighted random forests," Expert Systems with Applications, vol. 213, p. 118883, Mar. 2023, doi: https://doi.org/10.1016/j.eswa.2022.118883.

[6] "UCI Machine Learning Repository," Uci.edu, 2026. https://archive.ics.uci.edu/dataset/169/dorothea Accessed: Jan. 20, 2026.

[7] J. Kearns, *A Mechanism for Richer Representation of Videos for Children: Calibrating Calculated Entropy to Perceived Entropy*, Ph.D. dissertation, Univ. of North Texas, Denton, TX, USA, 2002.

[8] Scikit-Learn, "sklearn.ensemble.RandomForestClassifier." [Online] Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed: Jan. 20, 2026.

[9] Scikit-Learn, "sklearn.metrics." [Online], Available: https://scikit-learn.org/stable/api/sklearn.metrics.html. Accessed: Jan. 20, 2026.

[10] D. H. Seol, J. E. Choi, C. Y. Kim, and S. J. Hong, "Alleviating Class-Imbalance Data of Semiconductor Equipment Anomaly Detection Study," Electronics, vol. 12, no. 3, p. 585, Jan. 2023, doi: 10.3390/electronics12030585.